Rapid Generation and Utilization of Four-Dimensional Trajectories for Air Traffic Control and Management Applications in MACS

Al Globus*
San Jose State University, Moffett Field, CA, 94035

Richard H. Jacoby.†
Dell Services, Moffett Field, CA, 94035

Thomas Prevot‡
NASA Ames Research Center, Moffett Field, CA, 94035

James K. Wong§
Dell Services, Moffett Field, CA, 94035

The Multi-Aircraft Control System (MACS) enables evaluation of Next Generation Air Transportation System (NextGen) concepts with full-scale, real-time, human-in-the-loop simulation of complex air traffic control scenarios involving thousands of aircraft and large numbers of professional air traffic control personnel. MACS emulates current day controller displays and functionality, and implements a wide variety of advanced displays and controls including four-dimensional trajectories, interactive trajectory planning, automatic conflict detection and resolution, sector load histograms, weather probe and avoidance, automatic spacing, scheduling, and timelines. At the heart of these capabilities is software to rapidly and accurately generate and maintain trajectories. In this paper we describe MACS trajectories, how they are used, how they are generated and maintained, the different types of trajectories, MACS trajectory optimizations, and performance data.

Nomenclature

ADRS	=	Aeronautical Datalink and Radar Simulator
ANSP	=	Air Navigation Service Provider
ATC	=	Air Traffic Control
CDU	=	Control and Display Unit
ETA	=	Estimated Time of Arrival
FCS	=	Flight Control System
FMS	=	Flight Management System
LOS	=	Loss of Separation
MACS	=	Multi-Aircraft Control System
ms	=	Milliseconds
NextGen	=	Next Generation Air Transportation System
SID	=	Standard Instrument Departure
STA	=	Scheduled Time of Arrival
STAR	=	Standard Terminal Arrival Route
TRACON	=	Terminal Radar Approach CONtrol

---

* Senior Research Associate, NASA Ames Research Center Contract, MS 262-4, AIAA Member.
† Software Developer/Engineer, NASA Ames Research Center Contract, MS 262-3, AIAA Member.
‡ Research Engineer, Human Systems Integration Division, MS 262-4, AIAA Senior Member.
§ Software Developer/Engineer, NASA Ames Research Center Contract, MS 262-3.

I. Introduction

Over the next decades air traffic in the U.S. is expected to increase substantially (1) over the approximately 225,000 flights of all kinds per day in 2010[**]. The present air traffic control (ATC) system cannot accommodate such an increase (1). The NextGen system is being developed to enable safe and efficient operations as traffic increases. To evaluate NextGen concepts and automation, and fine tune them for operational use, a tool is needed that can conduct realistic, full-scale, real-time, human-in-the-loop, quantitative evaluation. MACS (2), the Multi-Aircraft Control System, has been repeatedly and successfully shown to be a tool well suited for this purpose.

MACS is 100% Java software used for distributed, full-scale, real-time, human-in-the-loop ATC simulation of thousands of aircraft and large numbers of operator workstations (3). Piloting and ATC can be automated or human-directed. MACS simulations are used to study human-computer, human-human and automation interactions of existing and proposed ATC operations, particularly potential NextGen capabilities. These studies help understand and quantify the safety and efficiency implications of different operational concepts. MACS can operate on a single computer, simulating and emulating all necessary functions, or on multiple computers communicating via the Aeronautical Datalink and Radar Simulator (ADRS) software (4). Typically, there is one computer for each controller or planner, one machine for each sector's aircraft controlled by a pseudo-pilot, and one each for simulation control and data collection. All MACS functionality is on every machine with an extensive configuration database, implemented as a standard file system, determining what portion of MACS capabilities are exercised on each machine.

A key component of MACS is rapid generation and utilization of four-dimensional aircraft trajectories. Trajectories are the four dimensional (latitude, longitude, altitude, and estimated time of arrival) paths aircraft take or are expected to take. They are critical to the correct and timely simulation of airspace operations. In this paper we:

1. describe MACS trajectories.
2. discuss most of the uses to which MACS trajectories are put.
3. describe each of the many types of trajectories MACS creates and their relationship to trajectories found in the real world, e.g., in flight deck computers.
4. describe how MACS trajectories are generated and maintained.
5. discuss some of the optimization techniques used to meet stringent real-time constraints for human-in-the-loop, full-scale, realistic simulations.
6. present performance data for trajectory-related MACS operations.

II. Trajectories

At the heart of MACS' capabilities is software to rapidly and accurately generate and maintain trajectories. A trajectory's four dimensional path is defined by a list of trajectory points. Aircraft fly, or are expected to fly, from trajectory point to trajectory point with turns as needed. Turns can originate before a trajectory point or after (fly over). Each trajectory point holds information about the location in all four dimensions plus other useful data such as course, length of the previous leg (segment from the previous trajectory point), required time of arrival, calibrated air speed, true air speed, ground speed, wind direction and speed, turn radius, start, center and end points, and the time the aircraft is expected to spend in the previous leg. In addition, there are special trajectory points; for example, holding trajectory points include the information necessary for an aircraft to circle while holding. MACS trajectories are derived from the flight state and/or lists of lateral trajectory points defined in various ways. Lateral trajectory points are associated with ground referenced location such as named waypoints[††] or latitudes and longitudes. These are specified in, for example, flight plans. Flight plans include the routes submitted by pilots or aircraft operators to Air Navigation Service Providers (ANSP) before takeoff. Additional trajectory points specify vertical and/or speed information; for example, top-of-descent. Such points are system-generated based upon along track distances.

[**] http://www.transtats.bts.gov/Data_Elements.aspx?Data=2 on 1 February 2011.

[††] Named waypoints are designated points on the ground published by the ANSP to simplify communication of ground positions between actors in the air traffic control system (e.g., controllers and pilots).

In addition, MACS trajectories maintain ancillary data. These include:

1. What led to the trajectory's creation, for example, human input or automation.
2. The next scheduled point. I.e., if the aircraft is being assigned a scheduled time of arrival (STA) by a scheduler it is stored with the trajectory, along with the scheduler that made the assignment. If there are multiple assignments the first one to be reached is stored.
3. The route, meaning the airports, runways and FMS (flight management system) procedures. An FMS procedure is a set of waypoints and speed and altitude restrictions at these waypoints that standardize various flight phases. Standard Instrument Departure (SIDS) and SID transitions are used to depart an airport. Standard Terminal Arrival Routes (STARS), STAR transitions, approaches and approach transitions are used to land at an airport. Jetways and Victor airways are used enroute, but MACS does not store these last two with trajectories, they are used by other parts of MACS. Note: the term 'transition' refers to the way an aircraft gets into or out of a specified procedure.
4. The profile, i.e., speeds and altitudes of flight phases such as departure, cruise, and approach.

The trajectory data structure is complex and requires significant effort and computation to create and maintain. However, it supports a wide variety of applications for ATC operations, simulation and emulation.

III. Trajectory Use

As trajectories are fundamental to air traffic operations, MACS uses trajectories for a wide variety of purposes. We will discuss some of them here.

Trajectories are used to control simulated aircraft motion. With the autopilot and FMS engaged, MACS aircraft will automatically change speed, pitch and bank to follow the current FMS trajectory, using the next trajectory point's information to determine desired speed, altitude, etc. This algorithm is not the same one used to calculate trajectory ETAs, so, as in the real world, trajectory ETAs are not exactly correct.

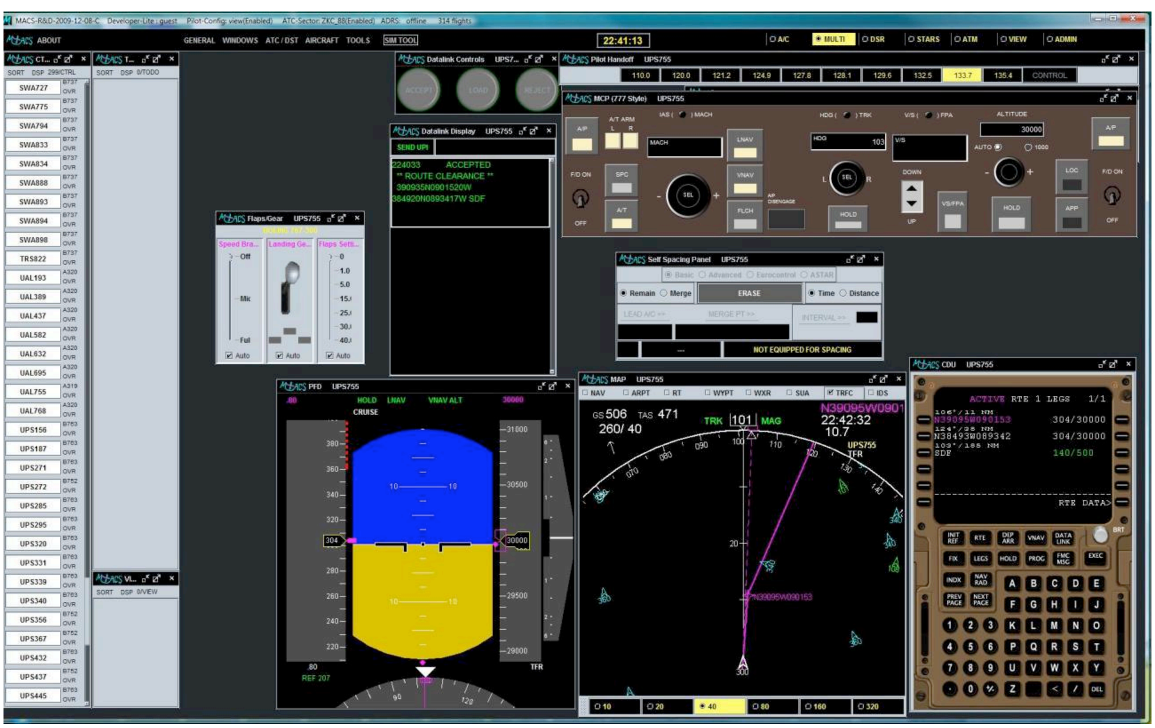

Figure 1. A MACS pseudo-pilot station display. *Note the MAP display in the bottom-right with a magenta trajectory.*

Trajectories are used on both the air and ground side to update displays informing pilots and controllers as to the paths an aircraft may take. Trajectories are color coded to indicate their function, e.g., flight plan vs. a planned trajectory. See Figure 2.

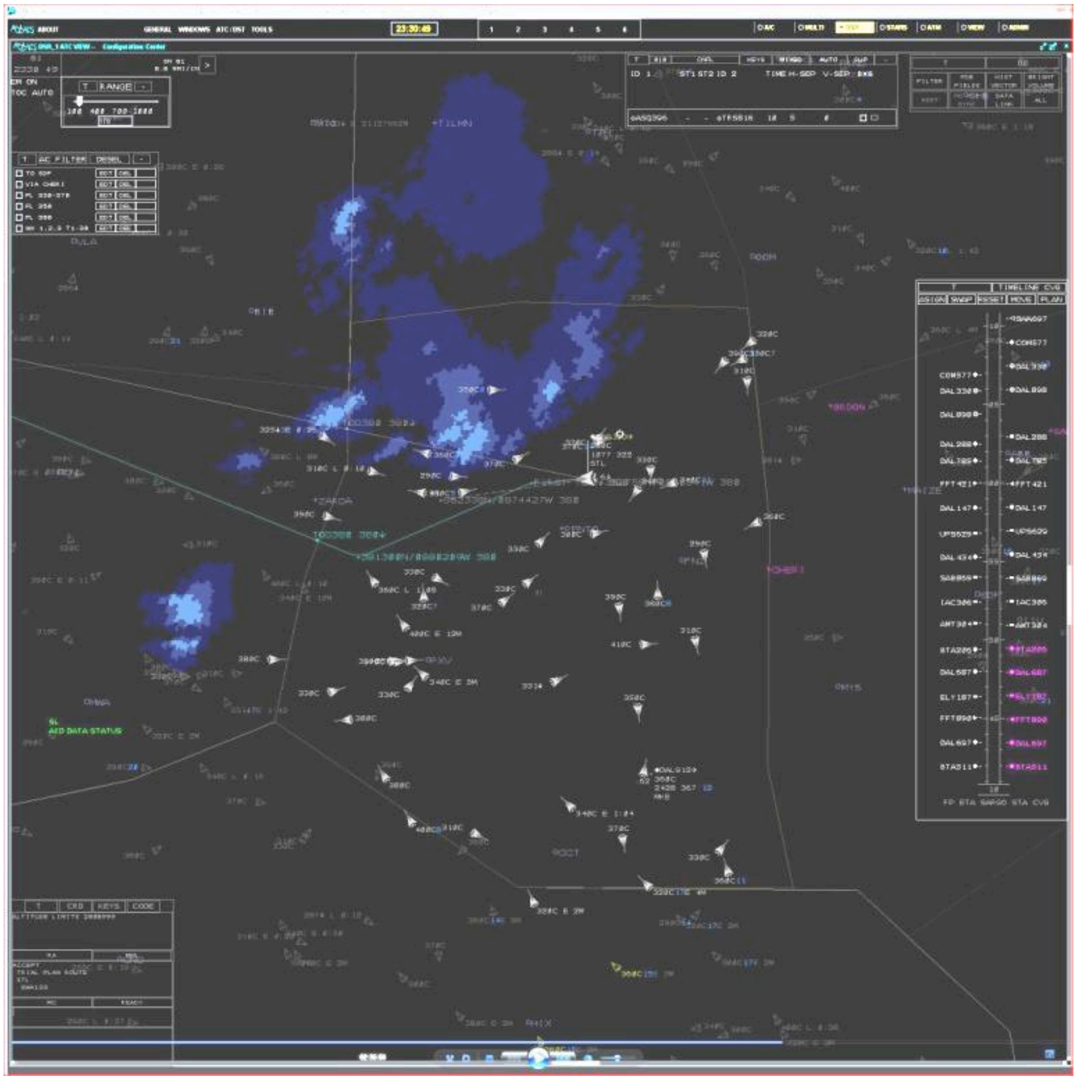

Figure 2. A MACS controller display. *Note the two trajectories near the middle. The aircraft's current trajectory, in grey, passes through weather. The planning trajectory, cyan, routes around the weather.*

MACS can be configured to indicate expected aircraft future motion by synchronized pulses moving along trajectories to aid visual aircraft-aircraft conflict detection. Pulses can be depicted as colored circles.

Controllers and planners can request planning trajectories for aircraft and directly manipulate them using mouse or trackball buttons to add and delete trajectory points. Altitude and speed changes can be entered from menus. Trajectory points can also be interactively dragged to search for conflict free trajectories, avoid convective weather, meet scheduled times of arrival or any other purpose. See Figure 3. If multiple trajectories share a trajectory point, they may all be dragged together. This is particularly useful for redirecting whole traffic flows around weather. MACS trajectories are fast enough for smooth trajectory display updates including real-time conflict graphics in typical operational conditions.

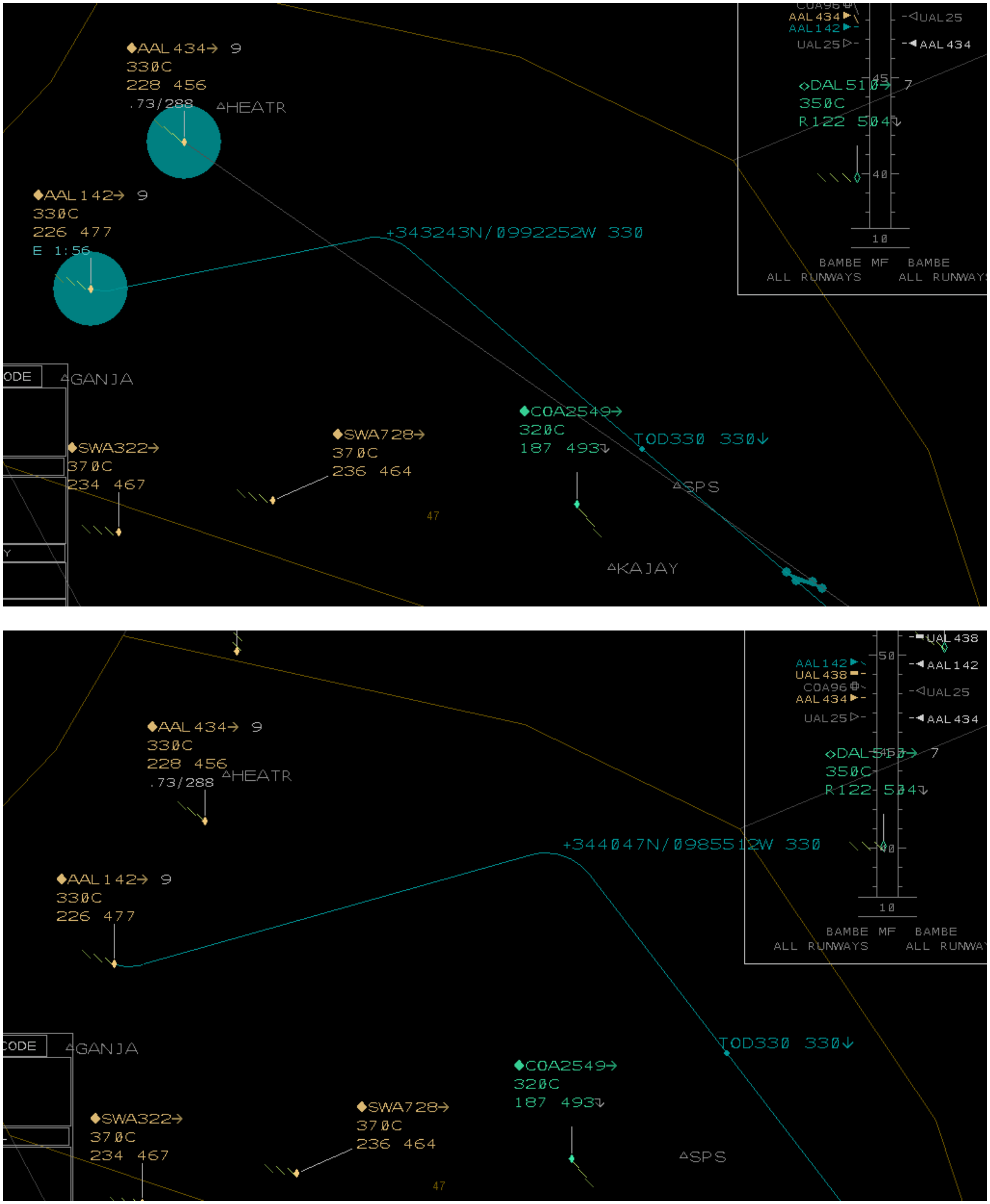

Figure 3. Trial planning tool. *The top image shows a planning trajectory (cyan) in conflict (the cyan circle around the aircraft symbol). By dragging the waypoint (bottom image) a conflict-free trajectory can be created.*

A. Conflict Functionality

Trajectories are also used to implement a conflict probe, which examines expected future aircraft locations to predict aircraft-aircraft conflicts. See figure 4. Controllers and/or automated software can then reroute aircraft to avoid loss of separation (LOS) events. An LOS is when two aircraft come closer than safety criteria allow, generally, in the en route airspace, 1,000 feet vertically and 5 nautical miles laterally. Conflict probe criteria are usually set more stringently (e.g., 7 nautical miles lateral separation) than what would constitute an LOS to avoid missed alerts.

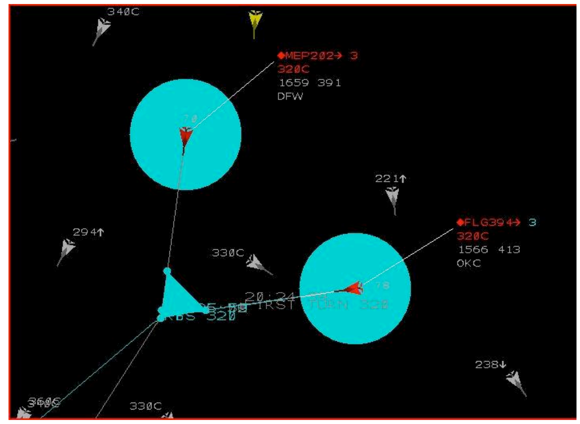

Figure 4. Conflict graphics. *The cyan filled circles indicate two aircraft that will lose separation if a planning trajectory is taken. The triangle indicates the portion of the trajectories where separation criteria will be violated.*

Once the conflict probe detects a conflict, graphics may be drawn on controller displays indicating the aircraft involved and the portion of the trajectories that are in conflict. Trajectory planning can then be used to resolve the conflict. Alternatively, automated software such as the AutoResolver Module (5) (6) can use fast trajectory generation to search the space of possible trajectories looking for an efficient conflict-free trajectory. The AutoResolver Module is conflict-free trajectory generating software developed by NASA Ames for the Advanced Airspace Concept Program and integrated into MACS.

Short-term conflicts are resolved with a conflict avoidance function that issues vectoring resolutions[‡‡] using the TSAFE (Tactical Separation Assisted Flight Environment) resolution algorithm (7). This functionality uses the conflict probe to detect conflicts but resolves these short-term conflicts with immediate turns without examining entire trajectories for secondary conflicts with other aircraft. After a vectoring resolution, aircraft are no longer on their flight plan. MACS trajectory capabilities and the AutoResolver Module can be used to automatically return such aircraft, once they have stabilized, to their flight plan via conflict-free trajectories.

Trajectories are also used to predict weather conflicts. Future aircraft positions are compared to current or predicted convective weather. The results can be displayed to the controller, e.g., time to weather, and or used by the AutoResolver Module to find a conflict-free path around the storm.

B. Traffic Load

Trajectories of all aircraft are examined to generate load tables and graphs. Load graphs are histogram-style depictions of the number of aircraft (and other measures) predicted to be in specific sectors or passing particular waypoints at intervals into the future. The availability of multiple trajectory types allows tuning the load displays; for example showing the effects of planned vs. existing flight plan trajectories and distinguishing between aircraft already airborne and aircraft still on the ground. See figure 5.

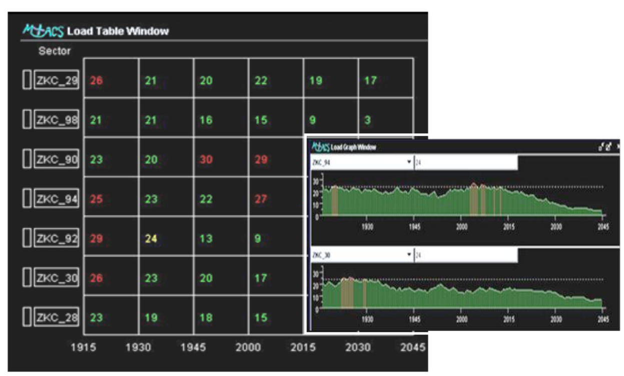

Figure 5. Load tables and graphs. *The table shows the number of aircraft projected by trajectories to be in various sectors in 15 minute time-slots into the future. Red numbers indicate over-loaded sectors. The histogram on the right shows similar information in graphical form at one minute intervals.*

[‡‡] A vectoring resolution is an aircraft turn to avoid a conflict.

C. Schedules and Timelines

Trajectories contain the ETA for each aircraft at each trajectory point. As a result, trajectories can be used to generate a list of the ETAs for aircraft arriving at a waypoint, runway, or airport. These data are used to generate timeline displays so controllers can examine the expected spacing of aircraft. Furthermore, controllers can interact with icons representing aircraft STAs to establish a schedule that avoids excessively close spacing and enhances efficient use of airspace and runways. See figure 6. The same information is used to implement meter lists.

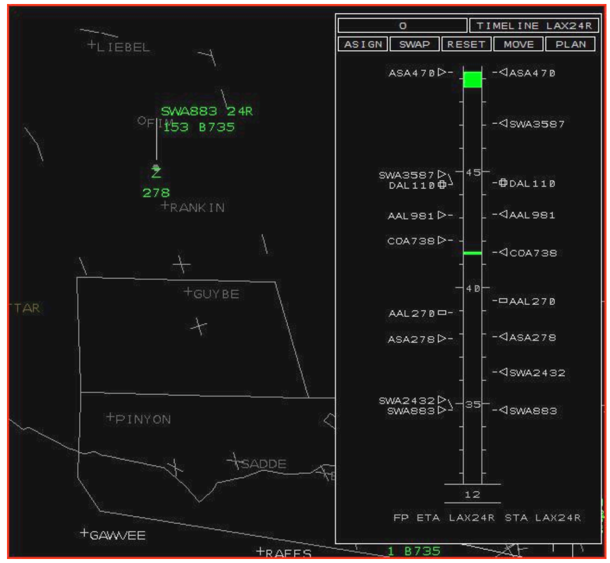

Figure 6. Timeline display. *The timeline on the right shows the estimated (left) and scheduled (right) times for aircraft to arrive at a runway. Aircraft lower on the timeline are in front of those above. The green rectangles indicate excess spacing in the schedule.*

In addition to controller generated STAs, MACS is capable of automatically generating STAs to spacing criteria using multiple scheduling algorithms. These schedulers use trajectory-supplied ETAs to compute STAs for aircraft time- or distance-based spacing values. These values are determined by applying a spacing matrix to each aircraft pair crossing a particular scheduling point. The matrix has entries for each possible pair of aircraft types as different sized aircraft have different spacing requirements (e.g., for wake vortex avoidance).

During interactive trial planning, timelines can display the planned ETA making it easy to match planned ETA to aircraft STA. This task can also be accomplished automatically using the AutoResolver Module.

D. Automation

The AutoResolver Module can be used to automatically generate conflict free trajectories which can then be data linked[§§] to appropriately equipped aircraft. The AutoResolver Module takes advantage of MACS' ability to rapidly generate trajectories and check them for conflicts to search the space of possible trajectories for the most efficient path to get an aircraft to its destination safely. The trajectory search space includes altitude changes, speed changes, and sets of two or more turns. The first turn is to avoid conflict, and at least one additional turn is needed to return to the flight plan. The space of altitude changes is small because final altitudes are always evenly divisible by 1,000 feet. The space of speed changes is severely limited by physics and aircraft type. The space of possible turns is quite large, but is discretized and uses patterns long established by controllers. As a result, a typical search may examine only 10-50 trajectories. The AutoResolver computes multiple options for both aircraft and then compares the conflict free options for efficiency (delay, path length). It also uses a two step process for conflict detection: a rough conflict check within the AutoResolver code and only the trajectories that pass that test are handed to the MACS trajectory generator/trial planner for detailed conflict probing.

Automated aircraft sequencing and spacing algorithms use trajectory information to predict future aircraft locations and make recommendations such as speed and altitude advisories. These advisories are primarily used for non-data link equipped aircraft and appear in the ATC display data block[***]. Calculating such advisories requires

---

[§§] Data link is the name of the method used to transfer digital information between aircraft and the ground, and between aircraft.

[***] The data block is typically several lines of text and text-like symbols associated with each aircraft on ATC displays. When data blocks are displayed there is typically a leader line drawn from the data block to the aircraft and the data block may be, and often is, positioned interactively by ATC personnel.

generating many trajectories to search the space of possible advisories for one that meets the spacing requirements. Some of this software is used internally by the AutoResolver Module to match STAs at waypoints.

E. Display

On MACS ATC displays aircraft appearance and data block contents are determined by a complex setup driven by various characteristics of the aircraft, including information determined by examining trajectories. For example, aircraft with immanent conflicts might be colored red. Data that control appearance include time to first conflict, time to weather entry, when an aircraft is in a given sector, when an aircraft crosses a line drawn by the controller, the altitude at which an aircraft will cross a waypoint within a time window, etc.

F. The Scenario Editor

MACS includes a scenario editor, a non-real time editing functionality used to set up simulations with the appropriate aircraft and weather. See figures 7 and 8. For thousands of aircraft and evolving weather, all designed to create specific problems for the controllers, this is a significant task. Trajectories are used to implement a time control so the experiment designer can instantly see the approximate location of each aircraft at any time during the proposed experiment, assuming the flight plan trajectory is followed exactly. Trajectories are also used for an automated function to create traffic load levels specified by the user. This facility uses the JavaGenes (8) search algorithm software to add and subtract aircraft to meet the desired criteria.

Figure 7. Aircraft, weather and trajectories displayed in the Scenario Editor. *The time control on the bottom right is set by the small blue triangle to display aircraft and weather at their positions at any time into the simulation.*

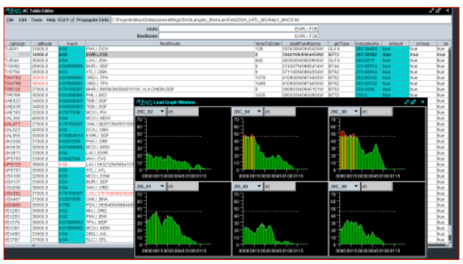

Figure 8. Aircraft and load tables in the Scenario Editor. *The load graphs in the center right bottom are the number of aircraft in various sectors in 15 minute time slots as projected by trajectories generated from flight plans. The table in the background contains all the initial conditions for each aircraft, one per row.*

G. Data Collection

Finally, experiments are of marginal value without quantitative data collection. Trajectories provide the data to compare the original flight plan with actual time and distance flown, trajectory change information and more. Trajectories are saved to disk periodically both for display and further analysis.

IV. MACS Trajectory Types

To support the uses of trajectories discussed above, MACS maintains a number of trajectory types on both the air and ground side. Each MACS workstation in a simulation calculates its own trajectories from partial information supplied by the initial scenario file, changes made by controllers and pilots, and compressed versions of trajectories communicated via data link between aircraft and aircraft and the ground. Thus, the same trajectory on different workstations may be slightly different as it may be calculated at different times with different input parameters. For example, a controller station may only have access to radar-based aircraft state and forecast wind information, while a pilot station has access to the actual aircraft state and wind information.

Trajectories are not communicated whole between the MACS ground and air-side. For aircraft equipped with data link, data link messages are exchanged, typically with limited information regarding trajectories. For other aircraft, verbal communication leads to pilot inputs to the flight control system which are used to generate trajectories. In both cases, even within a single workstation, air side and ground side trajectories are calculated separately, as in the real world, and may differ slightly. As previously noted, the ground side aircraft position can be subject to the noise and latency characteristics established by radar models, similar to the real world.

Each trajectory type has a separate thread to create and update the trajectory values, insuring that there is always a reasonably recent trajectory of each type available as needed. Each of these threads periodically performs two functions: 1) create or update trajectories and place them on a stack, replacing the trajectory for the same aircraft if already on the stack, 2) popping the stack to install the trajectories in the aircraft data structure. This two step procedure makes sure that only the latest trajectory of each type is installed. This is necessary because many different threads can create trajectories. For example, in addition to the thread associated with each trajectory type, the event thread creates trajectories in response to user inputs and the AutoResolver Module has its own thread for automatically separating aircraft. Each thread puts trajectories on the stack in a thread-safe manner.

All trajectories are ultimately derived from aircraft state and, in most cases, the flight plan. Some are generated from flight state alone. Flight plans are initially supplied for each aircraft in a scenario file that contain the aircraft information for a given experimental run. The flight plan for a simulated aircraft is much like a flight plan that might be submitted by a pilot in the real world. The flight plan is turned into an active FMS trajectory at the start of the simulation. During a simulation, pilots further modify the FMS trajectory as necessary. The flight plan is distributed to the ground system and modified by air traffic control personnel and ground-based automation.

Trajectories may be divided into air-side trajectories, those that might be found in the aircraft on-board computer either today or in the future, and those maintained and used by the ground-side (ATC) computers either today or in some future ATC system.

H. Air-side Trajectories

Air-side trajectories (see figure 9) that are created include:

1. Flight plan.
2. FMS active. This is the trajectory used by the autopilot and FMS, if engaged, to control the aircraft. On current commercial flight decks this is also referred to as the 'FMS planned' trajectory. It is depicted on the navigation display and referred to as the 'ACT RTE' in the Control and Display Unit (CDU).
3. FMS modified. This is a modified version of the FMS active trajectory maintained in the flight computer but not yet activated. On current commercial flight decks this is called the 'FMS modified' trajectory.
4. FMS planned. This is identical to the FMS active trajectory for ownship. For other aircraft and ATC applications it is the best estimate of what the FMS trajectory will be. It may be based on the planned route data linked up from the ground. It can also be a trajectory generated from the flight plan but not yet installed as the FMS active trajectory.
5. FCS commanded. This is the best estimate of how the aircraft will actually fly

Figure 9. Air-side FMS active and modified trajectories. *The magenta trajectory is the active trajectory. The cyan trajectory is a modified trajectory that goes directly to waypoint VHP.*

assuming no further input from the flight crew or the ground; i.e., what will happen if the pilot is completely hands off. It is very similar to the FMS active if the aircraft is on course vertically and horizontally, the auto-pilot is on and the FMS engaged. However, if the aircraft does not meet these conditions, the FCS commanded is the best approximation of where the aircraft will actually fly, which may not follow the flight plan. In certain applications this trajectory can be made available to other data link equipped aircraft for air-side conflict probing. The FCS commanded trajectory is also found on the ground side, but it is calculated separately and from different inputs that depend on the available information sources. This trajectory does not usually explicitly exist on commercial flight decks, but it has been proposed to be shared and used for conflict probing (9) and for improving situation awareness for the flight crew (10).

6. CSTRS (constrained). This is an unmodified version of the trajectory data linked up from the ground. It is not a full trajectory, but rather a set of constraints from which the FMS planned trajectory can be calculated.

Air-side trajectories that are references to other trajectories include:

7. Conflict. This is one of the other trajectories and is used for conflict detection. Usually it is the FCS commanded, but can be other trajectories in certain situations.
8. FMS priority. This is the FMS modified trajectory for the currently selected aircraft. In most experiments, individual pseudo-pilots control dozens of aircraft (or more). However, the state of only one aircraft is displayed at any one time, that of the currently selected aircraft. The FMS priory thread contains the FMS modified trajectory for only this one aircraft to insure that it is updated quickly.

I. Ground-side Trajectories

Ground-side (ATC) trajectories (see Figure 10) that are created for each aircraft include:

1. Flight plan.
2. FCS commanded. This is the best estimate of how an aircraft will actually fly. It is very similar to the flight plan if the aircraft is on course vertically and horizontally. However, if the aircraft is not on course in either dimension the FCS commanded trajectory will more closely approximate where the aircraft will fly. This is the same trajectory type as found on the air side, but on the ground it is calculated from the flight plan, controller inputs, and information data linked down from the aircraft, if available. It is normally used for the conflict trajectory.

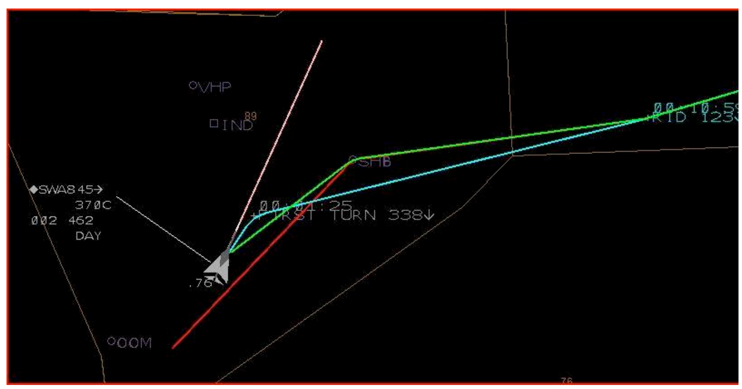

Figure 10. ATC trajectories. *The aircraft has flown off its route. Red is the flight plan. Pink is the FCS commanded. Cyan is the planning. Green is both the estimated and current.*

3. Nominal. This is a reference trajectory for aircraft arriving at an airport. In other words, this is the trajectory that *should* be flown to meet the scheduled arrival time at the runway or last scheduling point if no other instructions are received by the aircraft.
4. Planning. This is a trajectory generated on the ground either interactively by a controller or by automation; but not yet activated. The planning trajectory allows "what if" evaluation of possible trajectory options and is often referred to as the trail plan trajectory. The planning trajectory can be data linked to the flight deck of equipped aircraft and/or be sent to the ground system to amend the flight plan.
5. Estimated. When an aircraft is on course, this is the same as the flight plan. However, when an aircraft is off of the flight plan, for example, due to a TSAFE maneuver, this is a MACS generated trajectory that tries to rejoin the flight plan. Typically, this is the same as the initial planning trajectory provided to a controller for manipulation.
6. Pending. This is a planning trajectory sent to a controller by other ATC personnel, for example, a planner. The expectation is that this trajectory will be data linked or issued by voice to the aircraft but this has not yet occurred.

7. Proposed. This is the flight plan of a pre-departure aircraft. I.e., an aircraft that has yet to take off and is subject to ground delays.
8. Uplinked. This is the trajectory most recently data linked to the aircraft.
9. Advisory. This trajectory is generated to create controller advisories on how to meet a given constraint, usually an STA at a metering fix. A metering fix is a waypoint used to space aircraft to insure safe and efficient use of a runway. The advisories are recommendations for aircraft speed and altitude profiles. The trajectory generated from the advisories can be conflict probed.
10. fmsPlanned. This is the ground estimate of what the aircraft's FMS active trajectory must be. It is calculated from the flight plan, aircraft state, and controller instructions.
11. Reported. This is a trajectory data linked down from the aircraft in skeleton form and reconstructed on the ground. It is used when simulating of downlinked trajectory intent is required.
12. Current. This is intended to be the main trajectory used by the ground system. However, for a variety of reasons parts of the ground system will access other types. The current trajectory is derived from either the fmsPlanned trajectory (if available) or the estimated trajectory. Usually the differences are small.

Ground-side trajectories that are references to other trajectories include:

13. Conflict. One of the other trajectories that is to be used for conflict detection. It is usually the FCS commanded trajectory.
14. Priority. Similar to the air side FMS priority trajectory, this is for optimization purposes only. While a controller is dragging a trajectory point of planned trajectories these trajectories are put in a special thread to enable fast, smooth motion.

Each of the trajectory types has a specific role to play. Thus, each is at least a little different from the next and requires a somewhat different generation method.

V. Trajectory Generation

Trajectory generation is a complex process that varies somewhat between the different types of trajectories. As nearly all trajectory types ultimately depend on the flight plan, we will detail the process of converting a line of text containing airports, runways, waypoints and FMS procedures and other information into a flight plan trajectory.

1. The flight profile is created from controller inputs (on the ground side).
2. The text is parsed and converted into a list of lateral trajectory points. This process includes converting FMS procedures into a list of lateral trajectory points.
3. Altitude and speed restrictions from the FMS procedures are added to the appropriate trajectory points.
4. Altitude and speed restrictions are added to the last trajectory point.
5. Based on the aircraft position, the NEXT and ACTIVE trajectory points are determined. NEXT is the trajectory point after the aircraft, whereas ACTIVE is the first *lateral* point after the aircraft. At this step, these are the same as there are no vertical or speed trajectory points inserted, although lateral points may have speed or altitude restrictions associated with them. In some cases the ETA of the trajectory point before the NEXT trajectory point is set to the current time.
6. If there is only one trajectory point in the trajectory, a trajectory point is added for the current aircraft position.
7. Turn locations for each trajectory point are calculated.
8. Course and distance estimates for each trajectory point are calculated.
9. For descending aircraft, descent speed is updated for any speed restrictions.
10. Speed and altitude estimates for each trajectory point are calculated.
11. The wind speed and direction are determined for each trajectory point. This may be a function of time if the winds change during the course of a simulation. The course, true airspeed and ground speed are also estimated.
12. The outbound ground speed and true airspeed are calculated,
13. The calibrated airspeed is determined for each trajectory point.
14. Derived speeds: true airspeed and ground speed, are determined each point.
15. The distances from the last lateral point to each lateral point are calculated.

16. Altitude and speed trajectory points are added. For example, top-of-climb and begin-descent trajectory points.
17. The times between trajectory points are calculated based on leg length and ground speed.
18. The NEXT and ACTIVE points are determined again. This time, the NEXT waypoint may be a speed or altitude trajectory point but the ACTIVE point must still be a lateral trajectory point.
19. Estimated times of arrival are calculated for all trajectory points.

The flight plan trajectory is now complete, ready to be placed on the trajectory stack to be installed in the ground or airside version of the aircraft's data.

Different trajectory types sometimes require other processing. For example:

1. For some trajectories, a trajectory point representing the current aircraft position is entered into the trajectory. The leg to insert into is determined by heuristics every time the trajectory is created, so there is no dependence on history. This avoids problems when threads or workstations restart or are added to a simulation mid-stream. Planning trajectories must always start with an aircraft trajectory point.
2. ATC planning trajectories for aircraft not on the flight path must determine where to rejoin the flight plan. They also have a trajectory point placed two minutes off the aircraft nose since it will take time for turns to be sent to the aircraft and the pilot will require time to respond. Thus, the inserted trajectory point more accurately represents the path the aircraft will actually take than if the trajectory assumed an immediate turn at the time a planning trajectory was requested by ground personnel or automation.
3. ATC planning and FMS modified trajectories are modified by controller and pilot inputs, respectively.
4. FCS commanded trajectories are straightforward if the aircraft is on the flight plan. However, if the aircraft is not on course MACS must determine how the aircraft is likely to fly. If significantly off horizontally, this trajectory will be straight off the nose of the aircraft or curve towards the flight plan path. If off vertically or in violation of speed restrictions, the pilot commanded inputs are used heuristically to determine climb rate and acceleration or deceleration on the air side. Other data are used on the ground side.
5. The Nominal trajectory is determined from the flight plan but speeds and altitudes are determined by the preferred approach values to arrive at the runway at the aircraft's STA without reference to the aircraft's actual flight state.
6. Proposed trajectories have special trajectory points inserted at the time of presumed departure at the location of the airport or runway.

It should be noted that MACS' trajectory code is under constant development. This is a snapshot in early 2011 and will be less and less accurate as time goes on.

VI. Trajectory Maintenance

For every trajectory type that is created, MACS updates all or part of the trajectory by several criteria:

1. At low frequency (minutes) each trajectory is completely regenerated from scratch.
2. At high frequency (seconds) each trajectory has the ETAs and leg times in the trajectory points updated without adding or removing points.
3. Trajectories are modified in response to controller or pilot commands.
4. Trajectories are replaced by trajectories generated by the AutoResolver Module or the MACS spacing advisory code.
5. After a vectoring resolution to a conflict, trajectories are computed to return the aircraft to its flight plan trajectory.

High frequency updates only include ETAs and leg times because this operation is much faster than regenerating the entire trajectory. This is only one of the many trajectory optimizations MACS employs.

VII. Optimization

Trajectory generation and maintenance must be fast for three reasons:

1. There are up to 18 trajectories per aircraft and each workstation generates all the trajectories it uses. This means a workstation must maintain anywhere from hundreds to tens of thousands of trajectories.

2. Multiple trajectories must be manipulated interactively in real time. Not only must trajectories move smoothly as trajectory points are dragged with a track ball or mouse, but conflict graphics must appear and disappear on a sub-second time scale as planned trajectories enter and leave conflicts with other aircraft.
3. Each aircraft's trajectory must be checked against that of all other aircraft for conflicts (conflict probe) and to determine if the aircraft will encounter convective weather (weather probe). These functions are significantly sped up by a constant-time-interval mechanism.

To speed all operations, MACS employs an extreme multi-threaded architecture (up to 160 threads per workstation). Each separate function, including each window, has its own thread. For trajectory calculations this is taken further with a separate thread for each type of trajectory and additional threads for conflict probe, weather probe, automatic conflict resolution, timelines, scheduling, and more. This architecture permits MACS to trivially take advantage of modern multi-core processors, assign important interactive computations to their own threads, and even survive otherwise fatal intermittent errors by restarting threads that fail[†††].

More pedestrian, and also to speed up calculations, MACS converts all latitude/longitude to flat-Earth x,y around the center of the airspace being simulated. Only a few calculations require higher accuracy and these are special-cased.

This flexible and high-performance architecture enables a wide variety of trajectory applications. However, some operations are so compute intensive as to require additional speed up.

A. Conflict Probe

MACS contains a conflict probe to determine when two aircraft will violate separation criteria at some point in the future. The conflict probe is one of the few MACS operations that is $O(n^2)$ where n is the number of aircraft, each of which must be compared to all the others. As there may be thousands of aircraft, there can be millions of comparisons. Implementing conflict probe by directly examining the list of trajectory points would be prohibitively expensive, so MACS uses caching techniques to reduce calculation time. In addition, a number of heuristics are used to rapidly determine if two aircraft cannot possibly conflict.

For the conflict probe MACS calculates and caches each aircraft's future position in 15 second increments where the times for all aircraft are in sync. The length of these position lists is limited by the increasing unpredictability of an aircraft's location the further in the future one looks and there is no need to compare highly-uncertain future aircraft locations. Calculating aircraft position every 15 seconds in the future is done in the FCS commanded trajectory thread at three second intervals or when a planning or pending trajectory is created. Given two aircraft to compare for a conflict, the code simply walks the position lists simultaneously checking positions for a conflict.

B. Weather Probe

The weather probe is used to determine when aircraft are expected to encounter convective weather, i.e., storms. The implementation also creates a cache of future aircraft positions along trajectories, but at 60 second intervals as short encounters with the edges of weather systems are generally not a problem. Unlike the conflict probe, this facility is used by other parts of MACS so the range of future times for which this cache is needed is generally not known in advance. For this reason, the positions are calculated with a lazy evaluation. Furthermore, the class responsible for these calculations keeps track of the trajectory used and only recalculates fixed-interval cached points if the trajectory changes significantly. 'Significantly' is defined as a change in the number of trajectory points or more than a few seconds change in the ETA of the last trajectory point. Otherwise, only the cache points not already calculated in the selected time range are generated. As weather is represented by images positioned in the airspace, points not within any image can be ignored with a quick check. When a point falls within a weather image the appropriate pixel value is determined to decide the severity of the weather. All the weather entrance and exit data are kept in a data structure which is cleared when the weather moves or the trajectory changes significantly.

[†††] For example, rare thread timing errors can result in a null pointer reference. Such threads can be simply restarted with no loss of experimental realism. Thus, most important, an experimental run need not be restarted due to minor programming errors. This is crucial as professional controllers are used for experiments and they are available for only limited times. A stack trace is placed in the logs and the bug can be fixed later.

VIII. Measures of Conflict Probe and Trajectory Generation Performance

A timer class was developed and used to measure various aspects of MACS trajectory performance. For all measurements, MACS was run on a single Dell Precision M4400 laptop running Vista Service Pack 2. The laptop contained two Intel Core 2 Duo CPUs, T9800 at 2.93 GHz and four gigabytes of memory.

The first study compared the time spent calculating different types of trajectories in a scenario with 3,000 aircraft. See figure 11. About half of the aircraft began the simulation on the ground and became airborne over time. There was an average of about 1,400 in flight at any one time. Trajectories were generated for all aircraft, including those on the ground. There were no ATC or pilot inputs at all; the simulation was simply allowed to run. The conflict probe was turned off to reduce noise as the times recorded were wall clock times which are affected by other computations. Wall clock times are used because it is difficult to determine CPU times in MACS' extremely threaded architecture. All of the trajectory types that are directly calculated were included. Each trajectory type had a separate thread and each thread's primary method (periodicRunAction()) was executed no more often than every three seconds. Of course, in cases where an execution[‡‡‡] took more than three seconds the time could be longer.

Note that about half the trajectory types required very little time, the FCS commanded and FMS active trajectories dominated and the flight plan, ATC current, and ATC estimated required significant time as well. Also note that there are three columns for each trajectory type. The first, blue column is the average wall clock time for all executions during the simulation. However, as one of MACS' optimization techniques is to only update ETAs and leg times most executions, the vast majority of time is spent in those executions where the entire trajectory is regenerated. To capture these times the second two columns only average data points where time was greater than 10 or 100 ms respectively. Thus, while an average FMS active execution may take less than a second, complete trajectory regeneration for 3000 aircraft averages a little over 10 seconds, significantly greater than the nominal three second update rate.

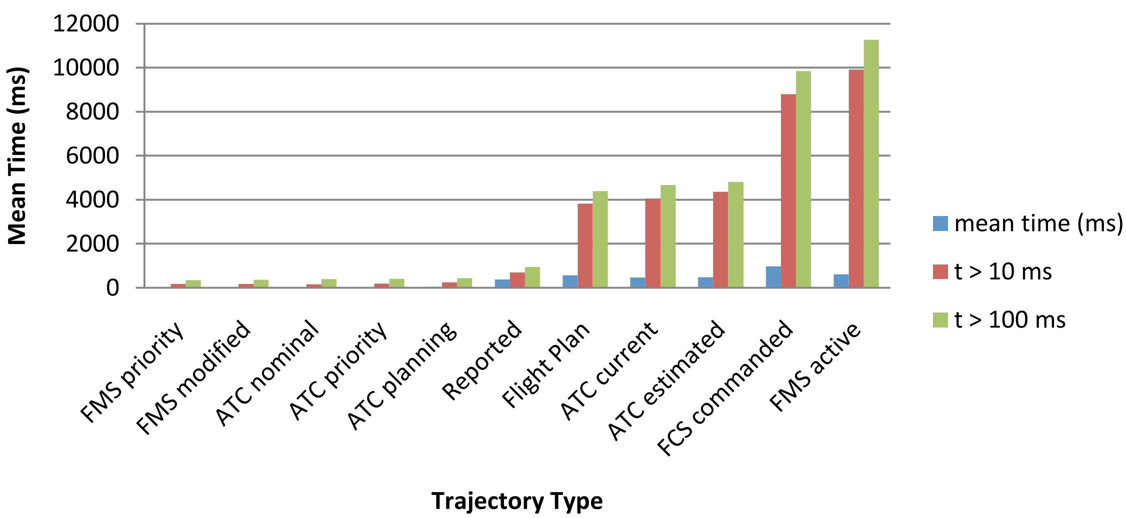

Figure 11. Mean trajectory time per execution. *Mean wall clock time in milliseconds per execution to examine and, if necessary, calculate different trajectories types for 3,000 aircraft. Since most trajectory calculation executions required very little time, the red and green data represent the mean for executions where the times were more than 10 and 100 ms respectively.*

[‡‡‡] An 'execution' refers to one call of periodicRunAction(), which processes all aircraft.

While Figure 11 shows the mean wall clock time for each execution for every type, Figure 12 shows wall clock times for each execution of the FMS active trajectory thread for the same run. Note the high peaks every ten executions or so.

Figure 13 shows conflict probe time as a function of the number of aircraft. The measure is mean wall clock time per execution. The red columns are time in milliseconds, the green columns the mean number of aircraft involved. The number of aircraft involved varies over the course of the simulation because only aircraft in the air are conflict probed, not aircraft on the ground. The horizontal axis is the percent aircraft used from a scenario with over 2,000 aircraft (including those on the ground). The five data points in this study were created by using a random sample of the aircraft in the study with the count being 100, 80, 60, 40, and 20% of the original aircraft. As one might expect from an $O(n^2)$ algorithm, the execution time decreases rapidly with the number of aircraft (ac in flight).

Figure 14 is a plot of conflict probe times where each execution is plotted. Note that every execution takes some time and there are occasional large peaks. This is driven by the changing geometry of the aircraft. Since none of the conflicts were resolved, the number of conflicts tends to grow over time.

Figure 15 is a plot of delay time when dragging a trajectory point during interactive trajectory planning. The horizontal axis has one data point for each mouse drag event, in temporal order. The vertical axis is wall clock time in milliseconds between the mouse drag event and completing the paintComponent() call that draws the graphics. This means that the time until the graphics actually change is understated, but actually knowing when the graphics are presented on the screen is very difficult. The data were

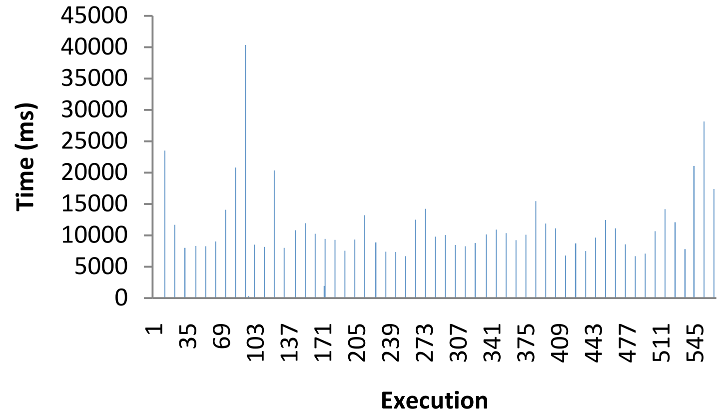

Figure 12. FMS active execution times. *Wall clock time in milliseconds per execution for FMS active trajectory generation for 3,000 aircraft. The horizontal axis is the execution number, which are at minimum three seconds apart. Note that most executions take zero time.*

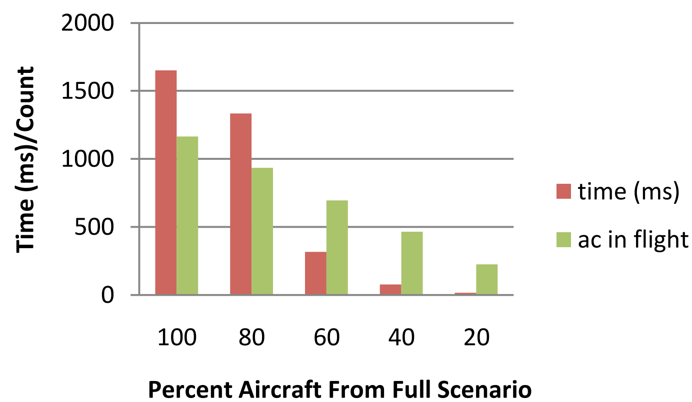

Figure 13. Mean conflict probe time. *The red column is the mean wall clock time for each conflict probe execution to complete. The green is the mean number of aircraft in flight during the simulation. The number varies as aircraft land and take off.*

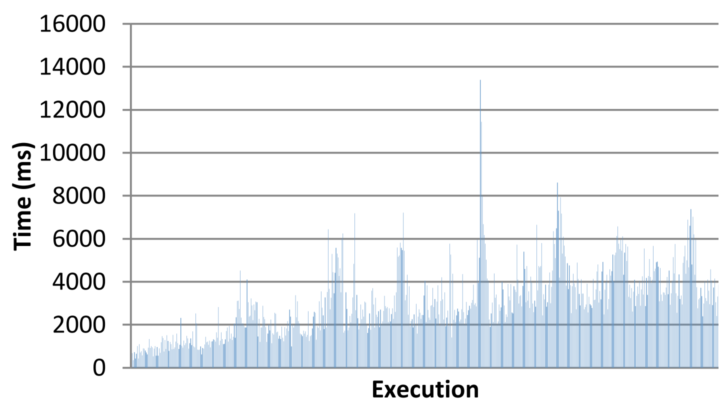

Figure 14. Conflict probe times. *Conflict probe wall clock time per execution.*

gathered on a scenario with over 1,000 aircraft by selecting four aircraft at random and trial planning them one at a time. Note that most of the time the graphics update within 200 ms, but near the end a situation developed where updates were significantly slower – almost a second in one case. This is because the amount of computation necessary is a strong function of the exact configuration of aircraft, where the trial plan is, and many other factors. Fortunately, in real operations not all aircraft are generally conflict probed, but rather only those in one sector or group of sectors at a time. Thus, the data in figure 15 exhibit much longer times than are generally found in experiment.

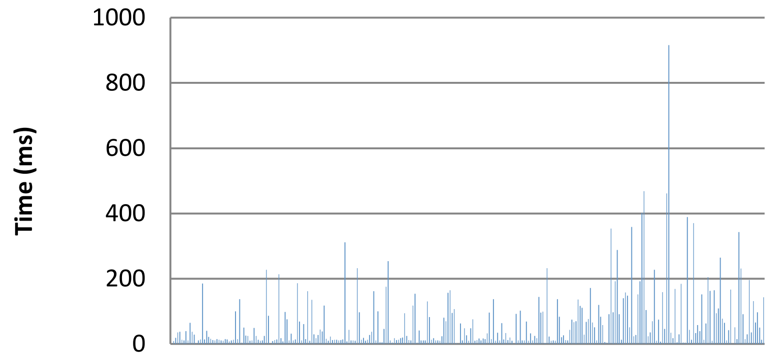

Figure 15. Trial planning delay time. *The vertical axis is a lower bound on the time between controller inputs (mouse or trackball move) and when the graphics update. There is one data point per mouse or trackball move event and the horizontal axis is ordered by time. 1,000 aircraft.*

IX. Conclusion

MACS simulations are used for real-time, full-scale, human-in-the-loop tests of human-computer and human-human interaction of NextGen concepts and automation essential to improvements in the nations' ATC system. As trajectories are at the heart of many modern ATC operational concepts, each MACS workstation generates and updates a number of trajectories for each aircraft. These are used to simulate aircraft motion and to implement a wide variety of tools and displays to implement both current-day and future ATC operational concepts. The MACS trajectory implementation is fast enough for interactive trajectory planning and conflict detection among thousands of aircraft. MACS' high-performance implementation and use of trajectories is a vital part of NextGen air traffic management research and automation prototype development and evaluation.

Acknowledgments

Current MACS development is funded under NASA's Airspace Systems program. We sincerely appreciate the help of many dedicated individuals at the NASA Ames Airspace Operations Laboratory, including Everett Palmer, Nancy Smith, Todd Callantine, and Paul Lee.

References

1. FAA Joint Planning and Development Office. *Concept of Operations for the Next Generation Air Transportation Systems, Version 2.0.* s.l. : FAA, 2007. http://www.jpdo.gov/library/NextGen_v2.0.pdf.
2. *Exploring the Many Perspectives of Distributed Air Traffic Management: The Multi Aircraft Control System MACS.* Prevot, Tom. Cambridge, MA : HCI-Aero, 2002. International Conference on Human-Computer Interaction in Aeronautics. http://humanfactors.arc.nasa.gov/ihi/research_groups/air-ground-integration/publication_papers/Pr2002-MACS.pdf.
3. *A Human-in-the-Loop Investigation of Multi-Sector Planning Operations in the NextGen Mid-Term Timeframe.* Smith, Nancy, et al. Fort-Worth, TX : AIAA, 2010. 10th AIAA Aviation Technology, Integration, and Operations Conference. http://humanfactors.arc.nasa.gov/publications/Smith-etal-ATIO-MSP2_final.pdf.
4. *A Multi-Fidelity Simulation Environment for Human-in-the-Loop Studies of Distributed Air Ground Traffic Management.* Prevot, Tom, et al. Monterey, CA : AIAA, 2002. AIAA Modeling and Simullation Conference and Exhibit. http://humanfactors.arc.nasa.gov/ihi/research_groups/air-ground-integration/publication_papers/Pr2002-MultiFidelitySim.pdf.

5. *Automated Conflict Resolution for Air Traffic Control.* Erzberger, Heinz. Hamburg, Germany : Congress of the Aeronautical Sciences, 2006. ICAS 2006 25th International Congress of the Aeronautical Sciences. http://airtrafficconflictresolutions.arc.nasa.gov/aac/ICAS2006_Erzberger.pdf.
6. *Automated Conflict Resolution, Arrival Management and Weather Avoidance for ATM.* Erzberger, Heinz, Lauderdale, Todd and Yung-Cheng, Chu. Nice, France : International Congress of the Aeronautical Sciences, 2010.
7. *Algorithm and Operational Concept for Solving Short Range Conflicts.* Erzberger, Heinz and Heere, Karen R. Anchorage, Alaska : International Congress of the Aeronautical Sciences 2008, 2008.
8. *JavaGenes: Evolving Molecular Force Field Parameters with Genetic Algorithm.* Globus, Al, Menon, Mahdu and Srivastava, Deepak. 5, 2002, Computer Modeling in Engineering and Science, Vol. 3, pp. 557-574. http://alglobus.net/NASAwork/#evolution.
9. *Development and Evaluation of an Airborne Separation Assurance System for Autonomous Aircraft Operations.* Barhydt, Richard, Palmer, Michael T and Eischeid, Todd M. Yokohama, Japan : 24th International Congress of the Aeronautical Sciences, 2001.
10. Hutchins, Edwin. *The Integrated Mode Management Interface, Final Report.* Moffett Field, CA : NASA Ames Research Center, 1996.
11. *A Multi-Fidelity Simulation Environment for Human-in-the-Loop Studies of Distributed Air Ground Traffic Management.* Prevot, Tom, et al. Monterey, CA : AIAA, 2002. AIAA Modeling and Simullation Conference and Exhibit.